

Introduction To tThe ASP+ Page Framework, Part 2

**Susan Warren
Program Manager
.NET Frameworks
Microsoft Corporation**

Microsoft®
PDC 2000
Professional Developers Conference

Microsoft®
.net

the defining

point

Agenda, Parts 1 & 2

■ Part 1

- Page Basics
- Server Control Basics
- Code Basics
- Control Families
- Pagelets

■ Part 2

- Input Validation
- Targeting Uplevel Browsers
- Data Binding
- Data Binding With Templates

Input Validation



Input Validation

- Rich, declarative validation
- Validation controls declared separately for input controls
- Extensible validation framework
 - *4-225 SA Zip Code Validator Demo*
- Supports validation on client and server
 - Automatically detects uplevel clients
 - Avoids roundtrips for uplevel clients

Validation Example

- Validation controls connected using **ControlToValidate** property

```
<asp:TextBox id=TextBox1 runat=server />  
<asp:RequiredFieldValidator id="Req1"  
    ControlToValidate="TextBox1"  
    Text="Required Field"  
    runat=server />
```


Validation Controls


- **Supported Controls:**
 - **<asp:RequiredFieldValidator>**
 - **<asp:RangeValidator>**
 - **<asp:CompareValidator>**
 - **<asp:RegularExpressionValidator>**
 - **<asp:CustomValidator>**
 - **<asp:ValidationSummary>**

Page.IsValid

- Indicates whether all validation controls on the page are valid
- Example:

```
Sub Submit_click (s As Object, e As EventArgs)  
    If Page.IsValid Then  
        Message.Text = "Page is valid!"  
    End If  
End Sub
```


Targeting Uplevel Clients

The background is a solid green color. In the lower-left quadrant, there are three large, semi-transparent green circles. Thin, light-green curved lines connect these circles, creating a network-like pattern that extends across the lower half of the slide.

Targeting Uplevel Clients

- Goal: Pages render more richly to uplevel clients, but work well in downlevel clients too
- Page developers can identify target client
 - <%@ Page ClientTarget="Uplevel" %>
 - <%@ Page ClientTarget="Downlevel" %>
 - <%@ Page ClientTarget="Auto" %>
- In the Beta1 timeframe:
 - “Uplevel” means IE4 and later
 - Only validation controls render uplevel

Request.Browser

- Provides more granular control of what is rendered to a given client
- Exposes specific client capabilities

```
If Request.Browser.Frames = True Then  
    'Do something special  
End If
```

Data Binding



ASP+ Page Data Binding

- Provides a simple way to bind Web UI elements to data
 - Enables clean separation of code from UI
- Supports binding to any datasource
 - DataSets, DataTables, DataReaders
 - Collections (Arrays, Hashtable, etc.)
 - XML

Data Binding Simple Lists

- **Example controls:**
 - **<asp:ListBox>, <asp:DropDownList>, <Select>**
 - **<asp:RadioButtonList>, <asp:CheckBoxList>**
- **Steps to data bind**
 - **Declare a list control**
 - **Optionally set DataValueField and DataTextField**
 - **Set its datasource**
 - **Call its DataBind() method**

Datasource Example

```
Function GetSampleData As DataView
Dim ds As DataSet
Dim MyConnection As SqlConnection
Dim MyCommand As SQLDataSetCommand
MyConnection = New SqlConnection
("server=localhost;uid=sa;pwd=;database=pubs")
MyCommand = New SQLDataSetCommand _
("select title from Titles", MyConnection)

ds = new DataSet()
MyCommand.FillDataSet(ds, "Titles")
GetSampleData = ds.Tables("Titles").DefaultView
End Function
```

List Binding Example

```
Sub Page_Load(s As Object, e As EventArgs)  
    myList.DataSource = GetSampleData()  
    myList.DataBind  
End Sub
```

```
<asp:radiobuttonlist id="mylist"  
    DataTextField="title"  
    DataValueField="title_id"  
    runat="server"/>
```

Data Binding DataGrid

- **Steps to bind all columns in the datasource**
 - Declare an `<asp:datagrid>`
 - Set its datasource
 - Call its `DataBind()` method
- **Binding to specific columns in the datasource**
 - Declare an `<asp:datagrid>`
 - Declare its `Columns` collection
 - Set its datasource
 - Call its `DataBind()` method

DataGrid Example 1

```
Sub Page_Load(s As Object, e As EventArgs)
    myDataGrid.DataSource = GetSampleData()
    myDataGrid.DataBind
End Sub
```

```
<asp:datagrid id=myDataGrid HeaderStyle-
    BackColor="skyblue" runat="server" />
```

DataGrid Example 2

```
<asp:datagrid id=myDataGrid HeaderStyle-  
    BackColor= "skyblue"  
    autogeneratecolumns=false runat=server>  
    <property name="Columns">  
        <asp:ButtonColumn Text=select  
            command="select"/>  
        <asp:BoundColumn HeaderText="Part  
            Number" DataField="title_id" />  
        <asp:BoundColumn HeaderText="Title"  
            DataField="title"/>  
    </property>  
</asp:datagrid>
```

Data Binding With Templates

The background is a solid green color. In the lower-left quadrant, there are three faint, semi-transparent green circles of varying sizes. Thin, light-green curved lines connect these circles, creating a subtle network or flow diagram.

What Are Templates?

- **Templates enable “lookless UI”**
 - **Customize structure - not just style**
 - **Controls can be used within templates**
 - **3rd party controls can expose templates**
- **Not limited to lists or data binding**
 - **No fixed set of templates; controls may expose any number of them**
- **4-428 Building ASP+ Server Controls #1**
- **4-429 Building ASP+ Server Controls #2**

Data Binding A Templated List

- **Steps to bind DataList**
 - **Declare an <asp:datalist>**
 - **Declare desired templates**
 - **Set its datasource**
 - **Call its DataBind() method**

DataList Binding Example

```
Sub Page_Load(s As Object, e As EventArgs)
    myDataList.DataSource = GetSampleData()
    myDataList.DataBind
End Sub
```

```
<asp:datalist id=myDataList runat=server>
    <template name="itemtemplate">
        Your price:
        <%# Container.DataItem("price") %>
        Description:
        <%# Container.DataItem("desc") %>
    </template>
</asp:datalist>
```

Templates Data Binding

- **Templates DataBinding Syntax:**
<%# any valid code expression %>
- **Code expression is evaluated in the language of the page**
- **Evaluated when DataBind() method is called on the templated list control**
- **Data Binding expression can reference any member of the page**

Container

- “Container” is an alias for the template’s containing control
- “DataItem” is an alias for the current row of the datasource
- **DataList examples:**

```
<%# Container.ItemIndex %>
```

```
<%# Container.DataItem("Name") %>
```

```
<%# Container.DataItem("Price") %>
```

Template Binding Example

```
<asp:datalist id=myDataList runat=server>  
    <template name="itemtemplate">  
        Quantity: <asp:TextBox id=qty Text=  
            '<%# Container.DataItem("price") %>'  
            runat=server/>  
        Product: <asp:Label id=qty Text=  
            '<%# Container.DataItem("name") %>'  
            runat=server/>  
    </template>  
</asp:datalist>
```


Summary

- **ASP+ data binding lets you bind any datasource**
- **Templates act like a “skin” for server controls**
- **Validation controls provide powerful, declarative input validation**

Related Sessions

- **Other PDC sessions recommended:**
 - ***4-428 Building ASP+ Server Controls #1***
 - ***4-429 Building ASP+ Server Controls #2***
 - ***1-431 Leveraging IE 5.5 From ASP+ Server Controls***
 - ***4-324 Targeting Mobile Devices w/ ASP+***
 - ***4-322 ASP+ Tips and Tricks***
 - ***4-334 ASP to ASP+ Migration***

References

- **“A Preview of ASP+” Wrox Press**
 - Chapters 3 & 4
- **ASP+ Quickstart in .NET Framework SDK**

Questions?



Where do **you** want to go today?

Microsoft

Backup

